이 곡 어때.

곡 장르와 태그를 기반으로
플레이리스트를 추천하는 시스템 개발

프로젝트 참여 인원

아래 총 4명의 새싹 교육생이 참여함

윤혜영
데이터 전처리 임무

조경희
콘텐츠 기반 필터링 임무

장윤식
모델 학습 임무

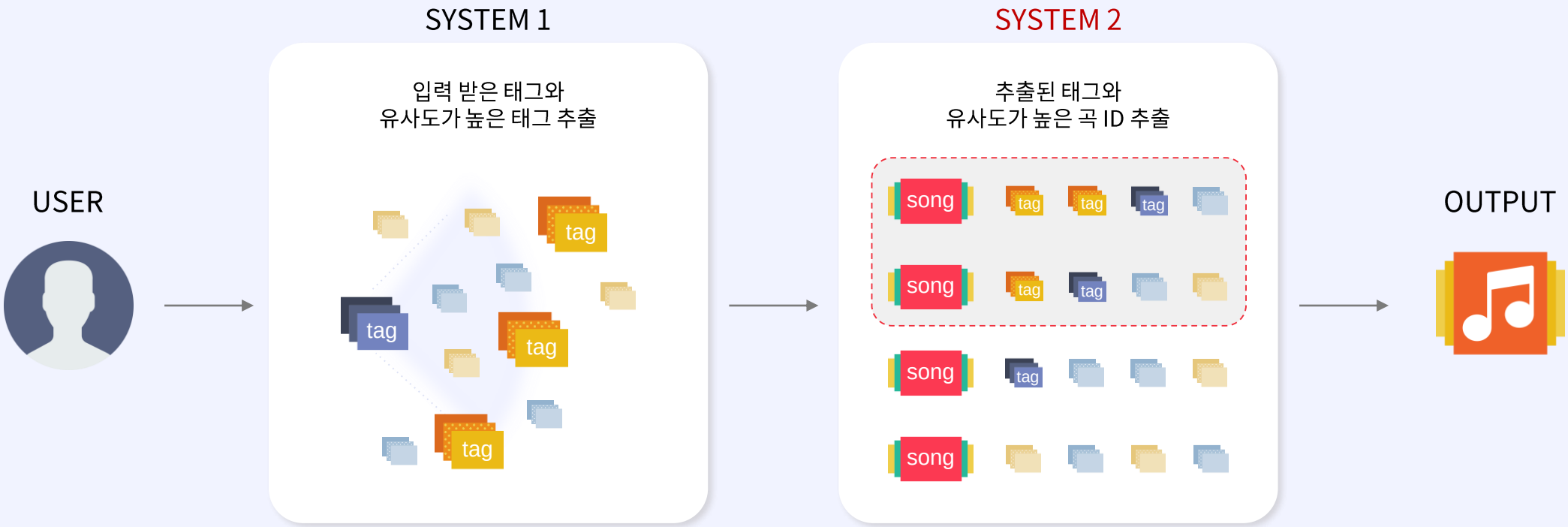
김 빈
자연어 처리 임무

기존 프로젝트 Blueprint

입력 받은 태그를 기반으로 유사도가 높은 태그를 추출하며, 이렇게 추출한 태그들을 기반으로 플레이리스트를 생성함
추출 태그를 기반으로 플레이리스트를 추천하는 경우, 이미 데이터 내 존재하는 플레이리스트만을 결과값으로 가질 수 있다는 한계가 있음

수정 프로젝트 Blueprint

입력 받은 태그를 기반으로 유사도가 높은 태그를 추출하며, 이렇게 추출한 태그들을 기반으로 곡 아이디를 추출하여 플레이리스트를 생성함



프로젝트 진행 절차

아래 절차는 Rutgers University에서 개발한 하이브리드 추천 시스템 개념도를 참고하여 작성했습니다.

1

데이터 수집

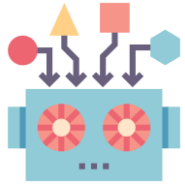
- 멜론 곡/아티스트 정보
- 멜론 곡 태그 정보

2

데이터 속성 추출

곡/아티스트별 속성 추출

3



모델 학습

Content-based Filtering
- Word2Vec, FastText
- feature별 유사도 분석
(코사인 유사도, 태그간 유사도)

4

결과 도출

태그/장르에 맞는 음악 추천
(태그로 분류되지 않은 단어를
입력하더라도 추천 음악 도출)

프로젝트 진행 절차 – 1) 데이터 수집 

카카오 아레나에서 다운받을 수 있는 멜론 데이터를 이용할 계획 – train.json, song_meta.json, genre_gn_all.json

train.json

- 플레이리스트 ID
- 플레이리스트 제목
- 태그 리스트
- 곡 리스트
- 좋아요 개수
- 수정 날짜

song_meta.json

- 곡 ID
- 앨범 ID
- 아티스트 ID 리스트
- 아티스트 리스트
- 곡 제목
- 곡 장르 리스트
- 곡 세부 장르 리스트
- 발매일

genre_gn_all.json

- 장르

프로젝트 진행 절차 – 1) 데이터 수집

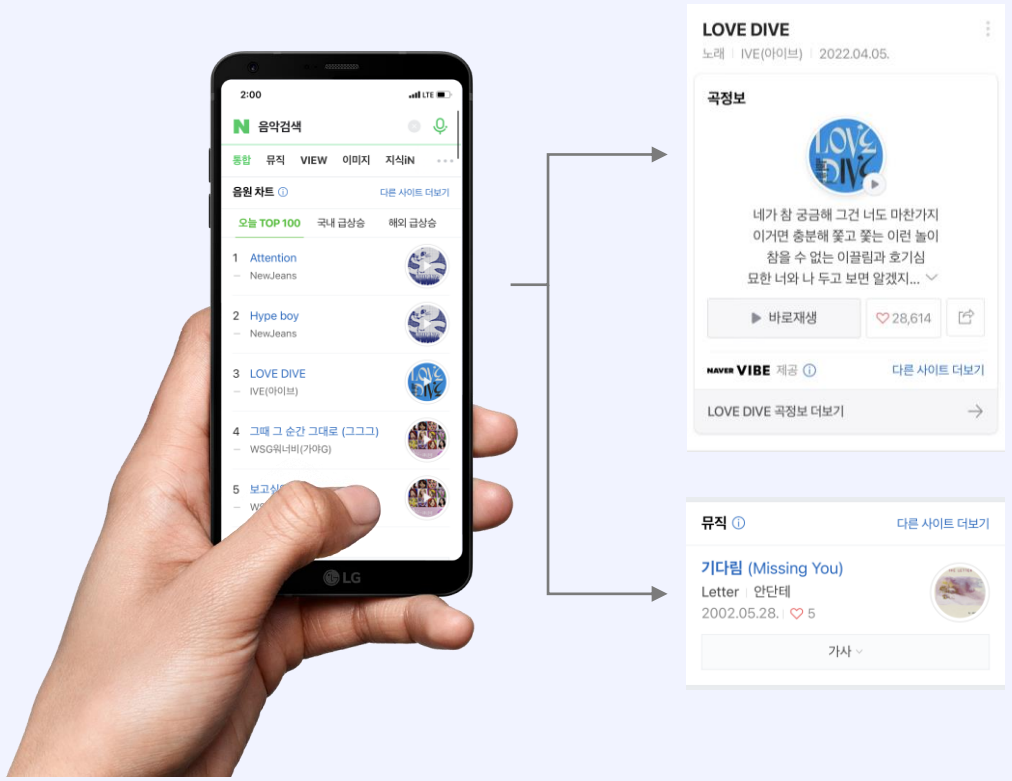
크롤링 함수는 아래와 같음

```
# url 만들기
base_url = 'https://search.naver.com/search.naver?query='

search_list = []

for i in tmp_list:
    search = i.replace(' ', '+')
    url = base_url + search
    search_list.append(url)

print(search_list)

## selenium으로 네이버 뉴스만 뽑아오기##
# 버전에 상관 없이 os에 설치된 크롬 브라우저 사용
driver = webdriver.Chrome(ChromeDriverManager().install())
driver.implicitly_wait(3)
driver.get(url)

# ConnectionError방지
headers = { "User-Agent": "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) Chrome/98.0.4758.102" }
```

```
dates = []

for i in search_list:
    driver.get(i)
    time.sleep(0.2) #대기시간 변경 가능

    original_html = requests.get(i,headers=headers)
    html = BeautifulSoup(original_html.text, "html.parser")

    # 발매일
    date = html.select("div.group_music > ul > li:nth-
child(1) > div.album_box > div > div.dsc_area > span:nth-
child(1) > time")

    # html태그제거
    pattern1 = '<[^>]*>'
    date = re.sub(pattern=pattern1,repl='',string=str(date))
    print(date)
    dates.append(date)

print(dates)
```

프로젝트 진행 절차 – 1) 데이터 수집

song_meta.json 발매일의 Null 값이 존재하는 경우가 있어서 크롤링을 통해 수집하여 채워 넣음




```
scraping_list.head()
```

	index	artist_name_basket	song_name	search	dates_another	dates	dates_f
0	562	Bryan Adams	Summer Of `69	Bryan Adams Summer Of `69		19850107	19850107
1	785	안단테	기다림 (Missing You)	안단테 기다림 (Missing You)		20020528	20020528
2	1543	노사연	만남	노사연 만남	20150507		20150507
3	1730	Cranberries	Promises	Cranberries Promises	19990101		19990101
4	2360	Deep Purple	Hard Lovin' Woman	Deep Purple Hard Lovin' Woman		19700901	19700901

두 가지 형태에 대한 크롤링 진행한 후, dates_another과 dates 컬럼 값을 비교하여 아래 규칙을 따르는 새로운 컬럼 dates_f를 생성함

1) dates_another 또는 dates 중 하나만 존재 → 값이 존재하는 컬럼 값 사용
2) dates_another과 dates 모두 존재 → 둘 중 더 과거의 값을 사용

프로젝트 진행 절차 – 2) 데이터 속성 추출

앞서 수집한 3가지 데이터를 결합 및 전처리하여 하나의 데이터셋을 구축한 후, 필요한 데이터 속성을 추출할 계획

train.json

+

song_meta.json

+

genre_gn_all.json

	tags	id	plylst_title	songs	like_cnt	updt_date	song_gn_gnr_basket	song_gn_dtl_gnr_basket
0	[락]	61281	여행같은 음악	[Hey Little Girl, Octagon, The Road, Honeymoon...	71	2013-12-19 18:36:19.000	[GN1400, GN1000, GN1100, GN1300, GN1900, GN0900]	[GN1001, GN1003, GN1901, GN1301, GN1101, GN140...
1	[추억, 회상]	10532	요즘 너 말야	[한사람을 위한 마음, Audition (Time2Rock), 기다리다, 좀 더 둘...	1	2014-12-02 16:19:42.000	[GN0100, GN1000, GN1800, GN0800, GN1700, GN260...	[GN0104, GN1502, GN1706, GN0103, GN2602, GN180...
2	[까페, 잔잔한]	76951	편하게, 잔잔하게 들을 수 있는 곡.-	[도시의 밤, I`m Alright, 너를 좋아하니까, 247 (Feat. AMJ)...	17	2017-08-28 07:09:34.000	[GN0100, GN0300, GN0800, GN1700, GN2600, GN040...	[GN0403, GN0401, GN0303, GN0301, GN0606, GN040...
3	[연말, 눈오는날, 캐럴, 분위기, 따듯한, 크리스마스캐럴, 겨울노래, 크리스마스,...	147456	크리스마스 분위기에 흠뻑 취하고 싶을때	[Into the Unknown (From "Frozen 2"/Sou...	33	2019-12-05 15:15:18.000	[GN0100, GN2200, GN1800, GN1300, GN0800, GN170...	[GN1706, GN0103, GN0506, GN0908, GN1506, GN180...
4	[댄스]	27616	추억의 노래 ㅋ	[눈물에 얼굴을 묻는다, 타인, 학원별곡 (學園別曲), 날 떠나지마, No.1, D...	9	2011-10-25 13:54:56.000	[GN0100, GN0300, GN2500, GN0200, GN0400, GN0600]	[GN0302, GN0403, GN0601, GN0103, GN2506, GN250...

프로젝트 진행 절차 – 2) 데이터 속성 추출

플레이리스트를 기준으로 태그를 붙이는 대신, 곡을 기준으로 태그를 붙이는 방식으로 변경

기존) 플레이리스트 기준 태그

	id	plylst_title	final_tags
0	112336	[헤어진 날] 꺼내 듣는 노래모음	[아픔, 발라드, '10-, 위로, 사랑, 슬픔, 감성, '00, 보컬 스타일, 랩...
1	61393	○ 여행&드라이브 차 안에서 함께들의면 기분좋아지는 리스트 ○	[기분, 록, 의, 발라드, '10-, 록/메탈, 인디음악, 힙합, 어반, 차안, ...
2	101019	# 추억의 댄스 가요 (운동, 드라이브)	['90, 발라드, 가요, 성인가요, 록/메탈, 랩 스타일, '00, 보컬 스타일,...

현재) 곡 기준 태그

	song_id	song_name	final_tags_chk
0	0	Feelings	[POP, 2000년대]
1	3	Feeling Right (Everything Is Nice) (Feat. Popcaan & Wale)	[Matoma, 일렉트로니카, 2000년대]
2	4	그남자 그여자	[픽, 2000년대, 프로, 아웃, 듀스, 데뷔, 발판, 시즌, 뉴에이지, Jude Law, 장만, 이지리스닝]
3	5	Para Los Enamorados	[재즈, 2000년대, Bye, Lupita, 겨울, 안녕, 소식]
4	6	Sibelius : Valse Triste Op.44 (시벨리우스 : 슬픈 왈츠 작품번호 44)	[San Francisco Symphony, 조깅, 운동, 2000년대, 관현악, 교향/관현악, 클래식, Herbert Blomstedt, 때]
...	...	...	...
615133	707984	Coffin For Head Of State	[월드뮤직, Fela Kuti, 90년대]
615134	707985	Change Of Heart	[POP, 80년대, Cyndi Lauper]
615135	707986	스치듯 안녕	[발라드, '10-, 윤종신, 2000년대]
615136	707987	숲의 빛	[컴필레이션, 뉴에이지, Nature Piano, 2000년대]
615137	707988	Queen 명곡 멜로디	[김경호, '90, 록/메탈, 90년대]

615138 rows × 3 columns

final_tags_chk는 (1) 곡 장르(대분류, 소분류) (2) 발행연도
(3) train_data 플레이리스트에 매핑된 태그 (4) 아티스트 정보를 통합하여 구성

프로젝트 진행 절차 – 3) 모델 학습

콘텐츠 기반 필터링의 방식을 이용해, 대상 자체의 특성을 바탕으로 추천 플레이리스트를 도출하는 모델 구축 예정

벡터 표현

곡을 표현할 수 있는 단어를
벡터로 변환하는
Word2Vec을 사용

유사도 분석

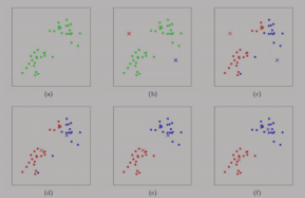
벡터의 유사도 측정을 위해
코사인 유사도를 사용

$$similarity = cos(\Theta) = \frac{A \cdot B}{||A||\,||B||}$$

$$= \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

군집화

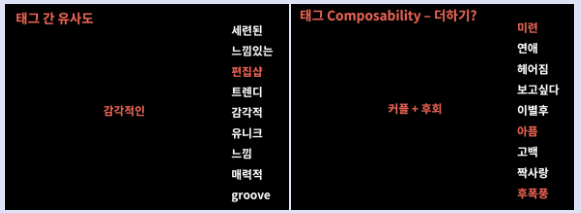
유사한 데이터의 군집화를 위해
K-mean clustering 사용



(참고: https://wikidocs.net/22660)

곡, 태그를 바탕으로 플레이리스트 추천

태그 간 유사도를 계산하여
유사한 태그를 도출할 수 있으며,
태그 간 연산(더하기, 빼기)을 통해
관련성 있는 태그를 확인할 수 있음



곡과 태그를 입력 받아
추천 플레이리스트를 도출하는 모델 구축

프로젝트 진행 절차 – 3) 모델 학습 및 추천 점수 계산

모델 학습 및 추천 점수 계산 시에는 아래 코드를 사용 – 평가 점수는 각 곡 태그 별 유사도의 평균과 같음

```
word2vec = Word2Vec(window=3,min_count=0,workers=5,sg
=1)
word2vec.build_vocab(tag_df.final_tags_chk.values)
word2vec.train(tag_df.final_tags_chk.values, total_examples
=len(tag_df.final_tags_chk.values), epochs=10)

word = input("입력해주세요 : ")
keyword = word2vec.wv.most_similar(word,topn=len(word2v
ec.wv.vocab))
keyword = dict(keyword)
keyword[word] = 1
```

```
>>> 입력해주세요 : 뉴에이지
```

```
def make_score(_dict,_list):
 temp = []
 error_count = 0
 for i in _list:
  temp.append(_dict[i])

 try:
  score = sum(temp)/(len(temp))
 except:
  score = 0

 return score


def apply_score(_dict,df):
 df['score'] = df.final_tags_chk.apply(lambda x : make_score(
_dict,x))
 sort_df = df.sort_values('score',ascending=False)
 return sort_df


test = apply_score(keyword,tag_df)
```

프로젝트 진행 절차 – 4) 결과 도출 🧠

입력한 태그 및 곡에 맞는 플레이리스트를 추천하며,
궁극적으로는 사전에 태그로 정의하지 않은 단어를 입력하더라도 추천 결과를 도출하는 모델을 목표로 하고 있음

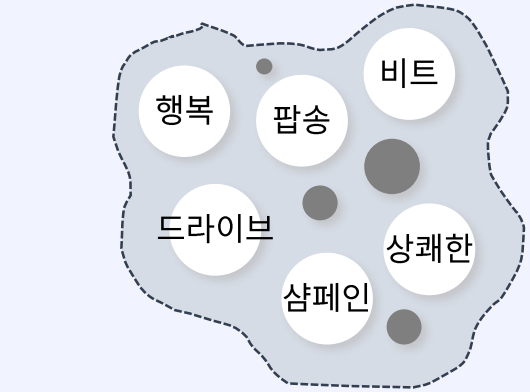
팝송 *Playlist* 2022 💔 나는 오늘 기분이 좋다 | 가볍고 시끄럽지 않은 느낌 🎵 *pop r&b mix* 🍃

조회수 3.3천회 • 12시간 전

오늘의 Chill Playlist

팝송 Playlist 2022 나는 오늘 기분이 좋다 | 가볍고 시끄럽지 않은 느낌 🎵 pop r&b mix TRACKLIST ...

새 동영상



Playlist 2022 💔 기분이 좋아지는 진짜 좋은 상쾌한 팝송 - 산뜻한노래, 행복 🎵 *pop music* ✨

343명 시청 중

오늘의 Chill Playlist

Playlist 2022 기분이 좋아지는 진짜 좋은 상쾌한 팝송 - 산뜻한노래, 행복 🎵 pop music ✨ TRACKLIST ...

실시간 새 동영상

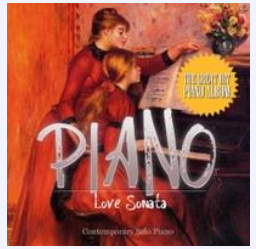
프로젝트 진행 절차 – 4) 결과 도출

‘뉴에이지’ 를 검색하여 점수가 높은 순으로 정렬한 결과는 아래와 같음

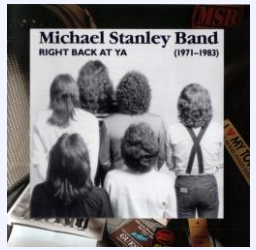
	song_id	song_name	final_tags_chk	score
419930	483222	Special To Me	[뉴에이지, 이지리스닝, 피아노, 2000년대]	0.688962
494273	568799	Just The Way You Are	[뉴에이지, 이지리스닝, 피아노, 2000년대]	0.688962
547130	629670	Piano	[뉴에이지, 이지리스닝, 피아노, 2000년대]	0.688962
472040	543172	If	[뉴에이지, 이지리스닝, 2000년대]	0.687285
342644	394202	Love Letter	[뉴에이지, 이지리스닝, 2000년대]	0.687285
...	...	...	...	...
300724	346118	불안	[MiRr, 2000년대]	0.034092
288427	331884	Crusade Of Darkness	[일렉트로니카, Ragnarok, 2000년대]	0.029516
9981	11553	Tennessee Whiskey	[Travis Lockwood, POP, 2000년대]	0.027430
544943	627143	Your Woman	[White Town, POP, 90년대]	0.025660
6374	7394	He Can`t Love You (Remastered)	[록/메탈, Michael Stanley Band, 2000년대]	0.020464

615138 rows × 4 columns

가장 유사한 노래
: Special To Me

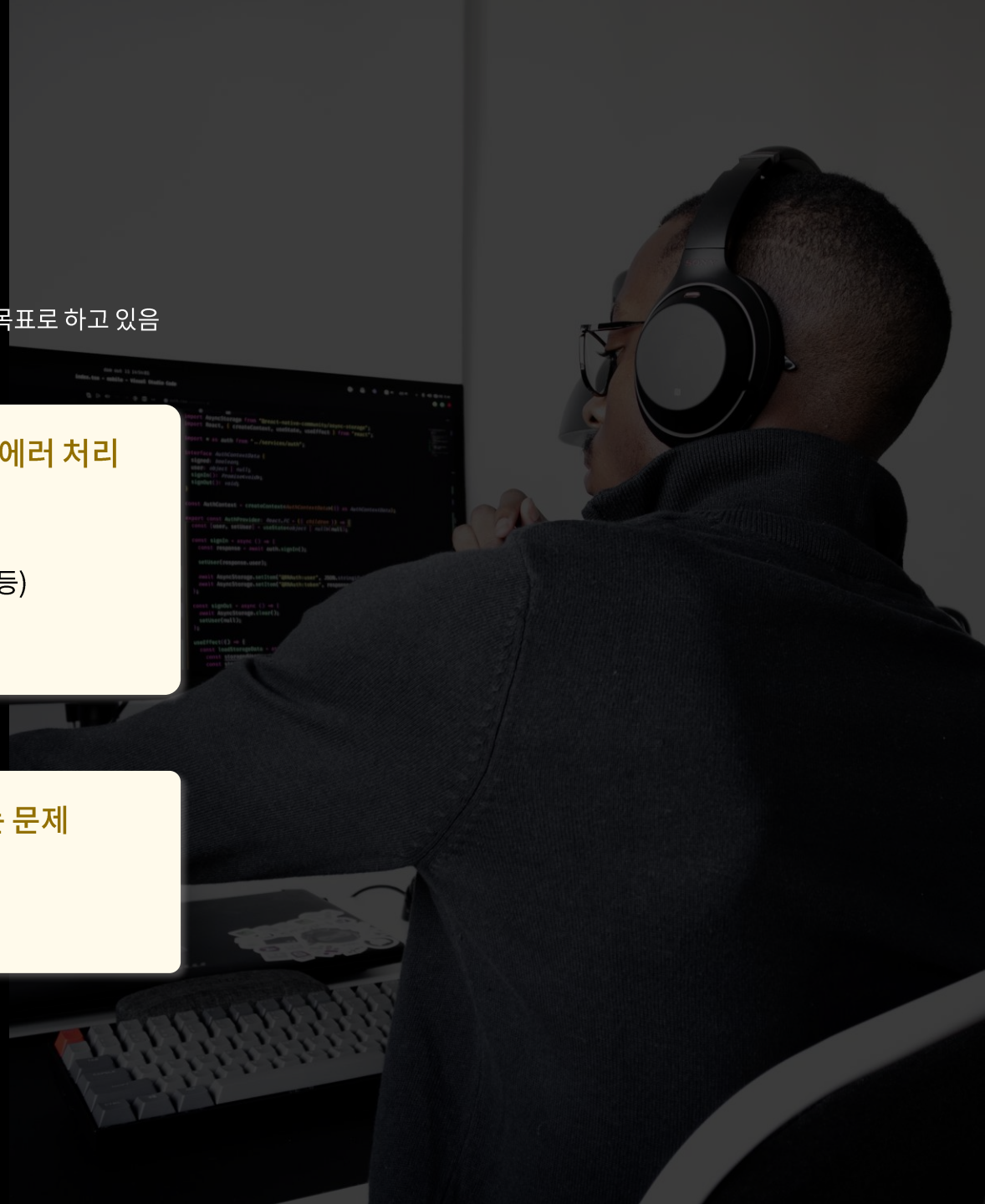


가장 유사하지 않은 노래
: He Can’t Love You (Remastered)



추후 프로젝트 진행 방향 Questions

입력한 태그 및 곡에 맞는 플레이리스트를 추천하며,
궁극적으로는 사전에 태그로 정의하지 않은 단어를 입력하더라도 추천 결과를 도출하는 모델을 목표로 하고 있음



학습된 말뭉치 내 존재하지 않는 태그를 사용자가 입력했을 경우 발생하는 에러 처리

Word2Vec의 학습 방법으로 인한 OOV(Out of Vocabulary) 문제가 발생할 수 있음

1) 새로운 단어에 대해서 가중치를 계산할 수 있는 모델 사용 (ex. TF-IDF, KoBERT 등)
2) 검색어를 입력 받는 대신 이미 보유하고 있는 태그를 제시하는 방식을 선택함
   - 웹 구현 시 추천 전 물어보는 질문에 대한 보기로 태그를 제시

큰 데이터 용량으로 인해 사용자 입력 후 결과값 도출에 오랜 시간이 걸리는 문제

→ 필요한 데이터만 추출하여 발표 1시간 전 해결!